A register file with 8.4GHz throughput for efficient instruction scheduling in a Pentium® 4 processor

Novat Nintunze and Giao Pham
Intel Corporation
Hillsboro, Oregon, USA
Phone: 001-503-613-9900
Fax: 001-503-613-5143
novat.nintunze@intel.com

Abstract

This paper describes a unique register file(RF) for ping-pong operation in 65nm CMOS process. The merged ping-pong reduces array width by 50%, doubles the frequency of access, and allows for same phase read and write. Implementation as a dependency matrix allows for all read wordlines to be asserted at once. A bypass scheme merged with the bitline contributes to a 27% leakage saving.

Keywords: microprocessor, scheduler, register file, bypass, performance, power.

Introduction

Scheduling micro-instructions (Uop) efficiently is a critical function in microprocessors. Scheduling tracks the valid bit of the Uop in data arrays along with the availability of physical source and destination, and checks the priority before dispatching the Uop to the execution unit. The Uop dependency matrix is a data array that contains a vector used to determine the readiness of the Uop to be dispatched. The matrix is coded as a chain of flip-flops, and "AND/OR" logic detects the dependency of the Uop. A typical implementation of the matrix creates routing with long interconnects, requires a large area and large devices with sub-optimal power/performance. This paper presents a design of a compact RF that encompasses the functionality of the flip-flop chain and collateral logic. Conventional fast RFs require a cycle for write_and_read [1, 2]. That's not fast enough for the dependency matrix because of the requirement for the resource to be accessible for write/read at every clock phase, and for all stored data to be read at the same time. Taking advantage of an architectural feature that guaranteed a single bit to be set high at any time, a same phase write-and-read RF was devised. This design in 1.2V 65nm CMOS logic technology[3], doubles the data throughput to 8.4GHz and reduces the array width by 50%. The bypass scheme integrated to the bitline pulldown results in 27% saving in bitline leakage at the burn-in supply of 1.4V.

Ping-Pong operation

The array design is of a "ping-pong" type: data is available every phase of the clock, and that increases the frequency of operation. In conventional RF where data write and read take a cycle, ping-pong implies duplicating the RF cells at the cost of area. In this paper, the phase1 and the phase2 parts of the RF are collapsed into a single RF, in a way that allows the data to be written and read from the same RF cells in the same phase. A fast latency, a smaller area and a simplified logic are achieved together. A pair of write and read ports is dedicated to an operation in one phase. Data written through each write port can be read by both read ports. "Ping" and "Pong" can be considered as 2 ports of the same memory cell. However, the operation of this RF can be extended to any number of ports.

Array organization

Fig.1. shows an 8-bit column of cell, with the unique memory cell that is used for both phases. Two write ports are used for each phase of data. In the figure, Wwl_A, Wdat_a, Rwl_A and BL_A are the phase1 write wordline, data, read wordline, and read bitline, respectively. Wwl_B, Wdat_B, Rwl_B and BL_B are the phase2 write wordline, data, read wordline and read bitline, respectively. A latch terminates the read path and merges the top and bottom halves.

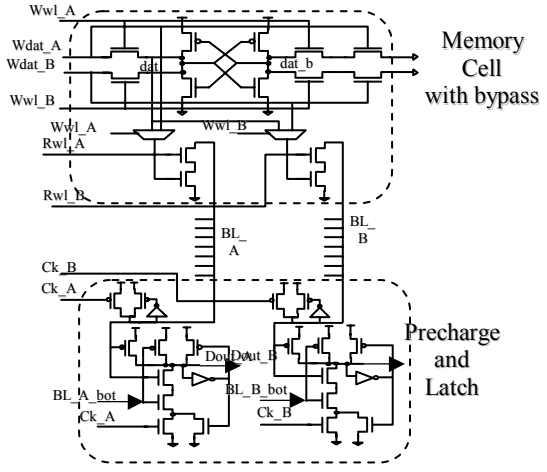

Figure 1: RF implementation: 8-bit column organization

The actual design in silicon is a 16x32 bit array, but there is no fundamental size limitation. To allow for

same phase write and read, a bypass mechanism is used: the data that is being written has a duplicate path that bypasses the memory cell and goes directly to the bitline. Since multiple read wordlines can be selected at once, and it is not possible to know a priori which data needs the bypass operation, the bypass logic cannot be implemented at the bitline level as usually done, but at the bit cell level. A mux switch that depends on the write operation taking place, selects whether the bitline contains the bit cell data or the write data. It is possible to operate as described because the architecture guarantees that a single bit per column is set high at any time.

This type of bypass is well suited to situations where all read wordlines are enabled together and there is a need to know the previously stored information as well as the present information, for example when scheduling an instruction. Figure 2 shows the signal relationship while reading and writing. The simulations were done at 1.2V supply, and 110C.

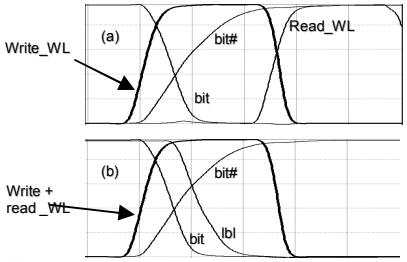

Figure 2. waveforms showing bypass operation.
(a) conventional case: read_WL has to wait for write to complete, (b) This design: read can happen at the same time as write.

The way data are written and read is a key difference between this RF and the conventional RF: this matrix array will write into cells in vertical direction while cells are read in horizontal direction and multiple read word-lines (Read WL 0..n) can be selected. The array layout for 1 port is shown in figure 3.

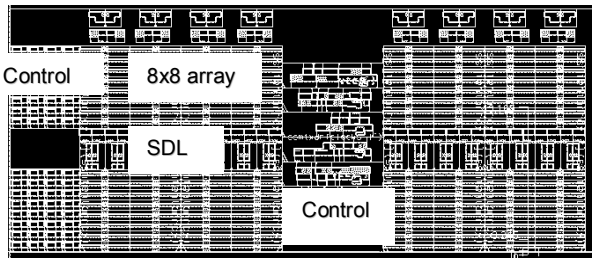

Figure 3. Register file layout showing half of the array

Leakage power saving scheme

Precharged bitlines have to hold their states, but with process scaling, they leak more and more. Device stacking provides the opportunity to decrease leakage. That was achieved as shown in figure 4. With this circuit, the design becomes more compact as well. The muxes are eliminated, but 4 devices are added to the read bitline pulldown path for data muxing, for a total gain of 20 devices per RF memory cell. That translates to 10K devices for 16x32 array. Alone, the bitline saving in leakage power total about 27% at burn-in. The elimination of the muxes removes about 5.2% of the dynamic power. At the array level, the total dynamic and leakage power saved, as measured by a power tool, reached 28%. In designs where susceptibility to noise is an issue, precharging of both sides of the bypass device is a possibility.

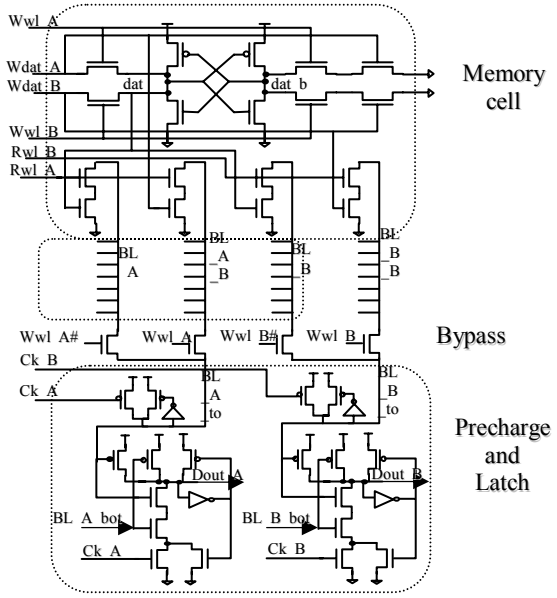

Figure 4. Bypass merged bitline

Conclusion

A register file for efficient dependency matrix implementation for a 65nm Intel Pentium® 4 processor was presented. The register file allows for write and read in the same phase and doubles the throughput frequency to 8.4GHz. Merged "ping" and "pong" circuits reduces the area and interconnect lengths by 50%. Merging the bypass in the bitline reduces the leakage power by 27% at burn-in and reduces the dynamic power by 5.2%.

Acknowledments: Our colleagues on the project and A. Farhang and F. Burke for review.

References

[1] Amit Agarwal, et al. 2004 VLSI circuits symposium Digest, pp.386-387.
[2] Michael Golden and Hamid Partovi, 1999 VLSI circuits symposium, pp.105-108
[3] P. Bai, et al. 2004 IEDM Tech. Digest, pp.657-660